Dynamical Adaptation in ORNs

Srinivas Gorur-Shandilya

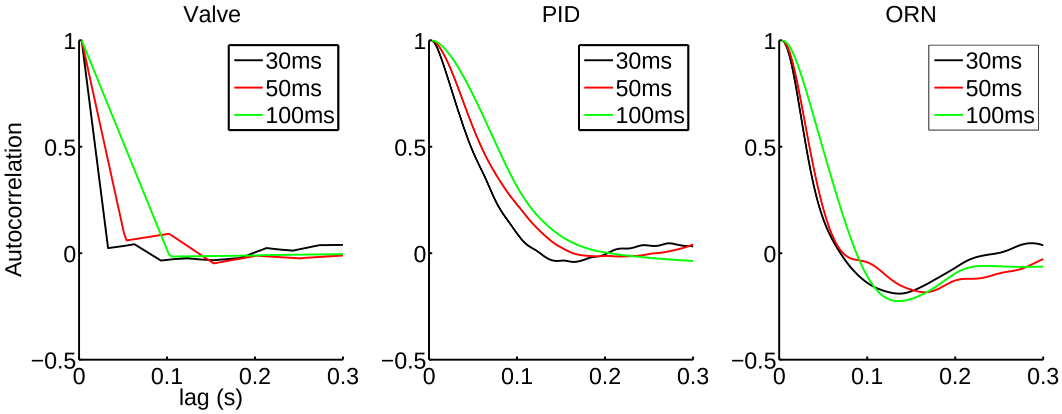
June 24, 2014

Contents

Effect of Correlation Time on Linear Filters

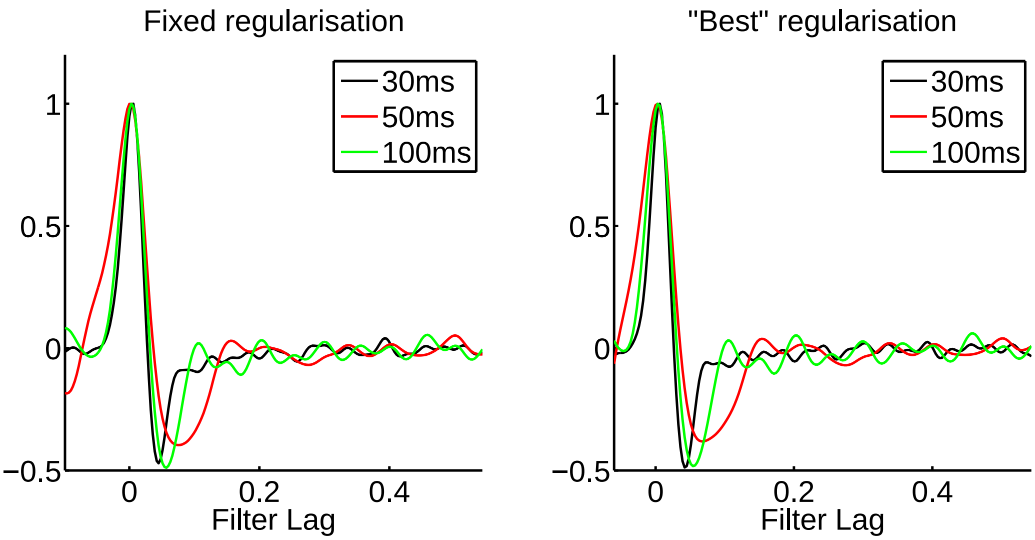
What is the effect of correlation time of the stimulus on the ORN response, and the filter that we calculate from this data?

First, we plot the autocorrelation functions of the valve, the stimulus and the ORN and check that they make sense.

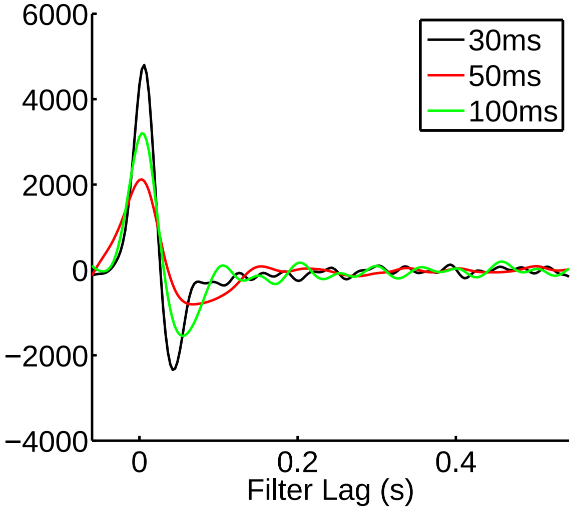


Are filters different for different correlations?

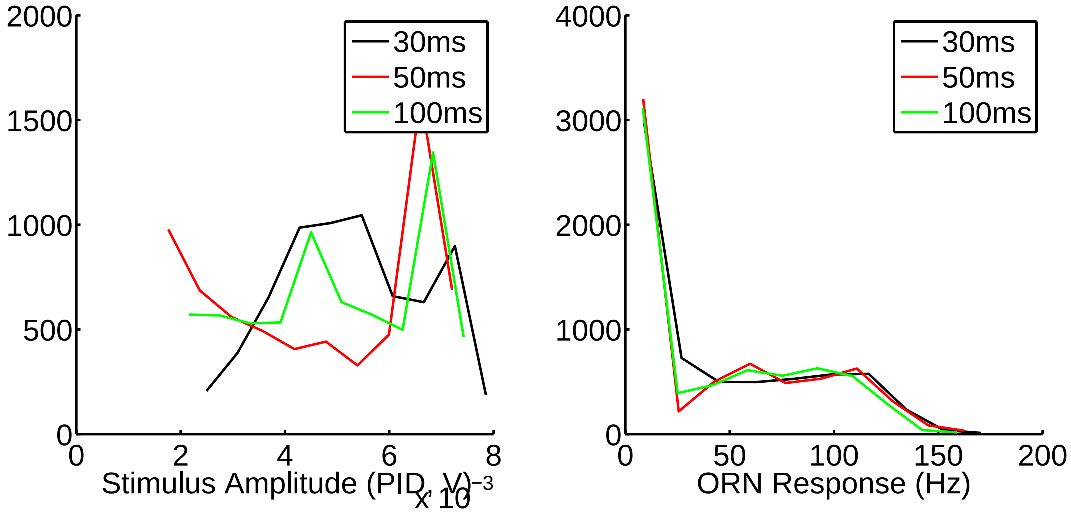
Now we fit filters to these data sets. In the figure below, the filters on the left are calculated using a fixed regularisation (of 1), and the filters on the right, the regularisation is varied in each case to find the "best" filter, minimising errors, high frequency components, and having a gain as close to unity as possible. In either case, we see the filters are slightly different for the three data sets; the correlation time of the stimulus seems to affect the filter we extract.



These filters differ not only in their shape but also their amplitude, which is a simple scaling because the input statistics are different.



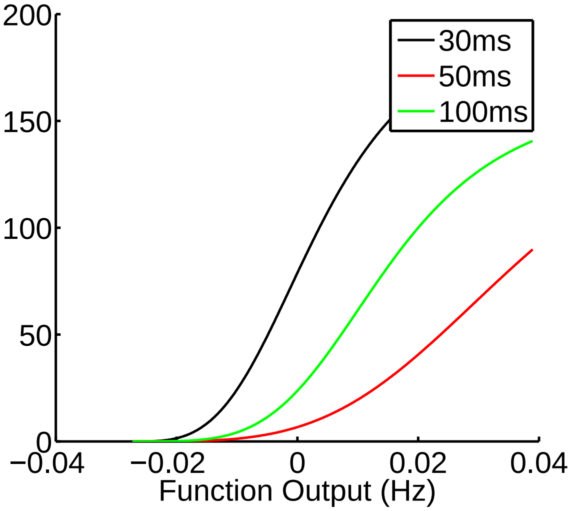
Why does the filter amplitude change so much? let's look at the stimulus and the response histograms:



Cross-prediction using different filters

To disentangle the effects of filter amplitude from possible changes in filter shape, we fit a static non-linearity to the output of each linear filter and use that to predict the output.

```

Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to
its initial value is less than the default value of the function tolerance.




Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to
its initial value is less than the default value of the function tolerance.




Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to
its initial value is less than the default value of the function tolerance.
```



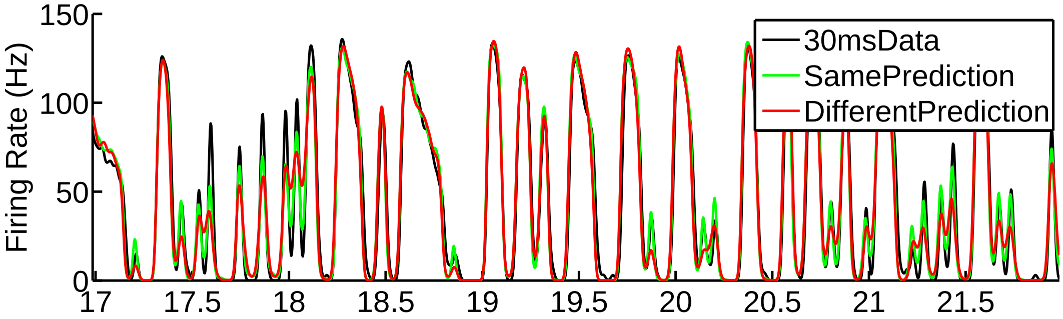
How good are these filters at estimating the data? In the figure we below, we show the two extreme cases (the 30ms and 100ms correlated data) and the predictions from the filters calculated from their own data sets, along with predictions from filters calculated using the others' data. The cross-prediction filter outputs are then passed through the corresponding static non-linearities,

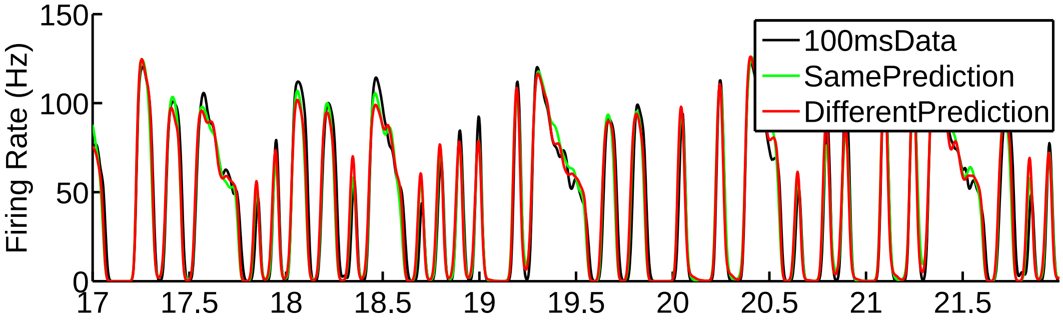
```
Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to
its initial value is less than the default value of the function tolerance.
```

```
Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to
its initial value is less than the default value of the function tolerance.
```

```
Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to
its initial value is less than the default value of the function tolerance.
```

```
Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to
its initial value is less than the default value of the function tolerance.
```





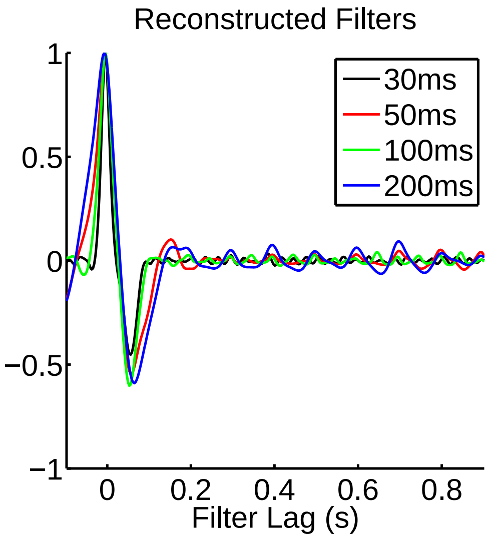
How good are these cross-predictions? For the three data sets we have, we measure how well the filter from each data set predicts the data. In the matrix below, the element in ith row and jth column shows the r-square of linear prediction by the ith filter to jth data set, where the first data set has 30ms correlated stimulus, the second has 50ms correlated stimulus, and the 3rd has 100ms correlated stimulus.

```
    0.9344    0.9273    0.9155
    0.8365    0.9486    0.8939
    0.9137    0.9363    0.9397
```

In particular, predictions using the fastest filter outperform predictions even from filters from the same data set.

Synthetic data: filter and non-linearity estimation

What effect does using stimuli with various correlation lengths have on our estimation of the filter? Here we use a ”fake” filter to represent the neuron, and pass stimuli with various correlation lengths to it, and then back out the filter from the synthetic data output.


Reconstructed Filters
1
0.5
0
−0.5
−1
0
0.2
0.4
0.6
0.8
Filter Lag (s)
30ms
50ms
100ms
200ms